

Citation for published version:

Thames, JL, Abler, R, Hyder, AC, Wellman Jr., RD & Schaefer, D 2011, Architectures and Design Methodologies for Scalable and Sustainable Remote Laboratory Infrastructures. in Azad, Auer & Judson (eds), *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines*. IGI Global, pp. 254-275. <https://doi.org/10.4018/978-1-61350-186-3.ch013>

DOI:

[10.4018/978-1-61350-186-3.ch013](https://doi.org/10.4018/978-1-61350-186-3.ch013)

Publication date:

2011

Document Version

Publisher's PDF, also known as Version of record

[Link to publication](#)

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines

Abul K.M. Azad
Northern Illinois University, USA

Michael E. Auer
Carinthia University of Applied Sciences, Austria

V. Judson Harward
Massachusetts Institute of Technology, USA

Managing Director:	Lindsay Johnston
Senior Editorial Director:	Heather Probst
Book Production Manager:	Sean Woznicki
Development Manager:	Joel Gamon
Development Editor:	Michael Killian
Acquisitions Editor:	Erika Gallagher
Typesetters:	Adrienne Freeland, Mackenzie Snader
Print Coordinator:	Jamie Snavelly
Cover Design:	Nick Newcomer, Greg Snader

Published in the United States of America by
 Engineering Science Reference (an imprint of IGI Global)
 701 E. Chocolate Avenue
 Hershey PA 17033
 Tel: 717-533-8845
 Fax: 717-533-8661
 E-mail: cust@igi-global.com
 Web site: <http://www.igi-global.com>

Copyright © 2012 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Internet accessible remote laboratories: scalable E-learning tools for engineering and science disciplines / Abul K.M. Azad, Michael E. Auer, and V. Judson Harward, editors.
 p. cm.

Summary: "This book presents current developments in the multidisciplinary creation of Internet accessible remote laboratories, offering perspectives on teaching with online laboratories, pedagogical design, system architectures for remote laboratories, future trends, and policy issues in the use of remote laboratories"— Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-61350-186-3 (hardcover) — ISBN 978-1-61350-188-7 (print & perpetual access) — ISBN 978-1-61350-187-0 (ebook) 1. Science—Study and teaching. 2. Laboratories—Computer-aided design. 3. Internet in education. I. Azad, A. K. M. (Abul K. M.) II. Auer, Michael E., 1948- III. Harward, V. Judson, 1951-

Q182.7.I58 2012

507.8'5—dc23

2011031133

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 13

Architectures and Design Methodologies for Scalable and Sustainable Remote Laboratory Infrastructures

J. Lane Thames

Georgia Institute of Technology, USA

Randal Abler

Georgia Institute of Technology, USA

Andrew Hyder

Georgia Institute of Technology, USA

Robert Wellman

Georgia Institute of Technology, USA

Dirk Schaefer

Georgia Institute of Technology, USA

ABSTRACT

With the increasing demand for distance learning opportunities in the higher education sector, there is an ever-growing need for the design and deployment of remote laboratories, especially for engineering, science, and technology curricula. In order to accommodate the offering of entire degrees for distance learning students whose curricula require remote laboratories, scalable information technology infrastructures that support the large scale use and deployment of these remote laboratories must exist. This chapter provides a discussion of architectures and design methodologies using technology such as command and control communications, Web 2.0, and cloud computing, which provide a scalable, manageable, and sustainable technological infrastructure-basis for large scale remote laboratory deployment.

INTRODUCTION

The last decade has given rise to many advances in Internet-enabled technology, and as a result of the “ubiquitous” Internet, the education sector has seen significant growth in the area of distributed education and distance learning. This growth is a result of institutions offering online coursework and providing remote campuses. Even though many universities are offering online coursework and distance learning, with approximately eighty-five percent of US universities considering distance learning and online coursework as vital components of their educational strategies (Kirkland, 2008), there is a significant challenge faced when offering distance learning coursework in the engineering, science, and technology sector. This challenge is the ability to offer coursework that requires laboratory instruction. As a result of this challenge, much research in the area of remote laboratories (RL) is being conducted. We define RL as the following: a remote lab, in general, includes an apparatus that is operated through a computer, possibly interfaced with audio and video equipment for real-time feedback to the remote user, and is controllable from a remote location via computer networks. Further, a remote lab incorporates laboratory units (the apparatus) that can, in certain cases, be moved, seamlessly, between different locations and computer networks. The apparatus in RL is any networked device that implements a remote lab environment. For the rest of this chapter, we will refer to the apparatus as a remote laboratory component (RLC).

One can enumerate several reasons for deploying distance learning curricula and associated remote laboratories, such as:

- Universities can reach out to students who, without distance learning technology, would otherwise not be able to obtain higher education and advanced degrees.
- Students can experiment with different configuration settings, get results very

quickly, and hence are encouraged to do more “what if” exploring than they would do in a traditional laboratory.

- Students have the flexibility to log in, conduct an experiment and complete associated assignments from any place in the world and at any time they choose.
- Remote laboratories provide broader access to expensive and/or specialized equipment and thus foster the concept of “magnet schools”.
- Remote laboratories help prepare students for the workplace of tomorrow, in which remote work and mass collaboration (Rippel, 2009) will be the norm.

This chapter provides discussion on the idea of remote laboratory infrastructures. The next section provides background material and discussion on previous work in the design of remote laboratories. Then, the chapter provides a discussion of requirements and characteristics of remote laboratory infrastructures; it introduces a systems model that provides a guiding framework for future developments of remote lab infrastructure, and the details the operation of a remote lab infrastructure using a case study narration. The chapter then closes with discussion of future trends and conclusions.

BACKGROUND

Since the mid 1990s, numerous works have been introduced in the design of remote laboratories. A few recent works have focused on software architectures for remote laboratories. A laboratory that supported a variety of remotely operated laboratory exercises in control systems and chemical, environmental, and mechanical engineering was developed at the University of Tennessee at Chattanooga (Henry, 2000). A remotely controlled physics experiment to determine the speed of light was developed by Enlo et al. (1999). Experiments involving semiconductor characterization were

developed by Shen et al. (1999). Hamza et al. (2000) developed a prototype remote laboratory system; their initiative led to the development of the Florida Atlantic University CADET (Center for the Advancement of Distance Education Technologies). They developed proof of concept prototypes and claim to have laboratories that are under development, including Electrical Element Characterization (for Electrical Engineering), Logic Design (for Computer Engineering), Motion and Friction (for Mechanical Engineering) and Metallic Elasticity (for Physics and materials in engineering).

One of the first comprehensive surveys on online higher education was published by Sloan-C and the Sloan Center for Online Education in 2004 (Sloan-C, 2004). At that time, their main finding was that the Associates degree granting institutions have the largest number of students taking at least one online course, representing about half of all the students studying online, while they were followed, in order, by Masters, Doctoral/Research, Specialized and Baccalaureate institutions with the smallest number.

Based on the Sloan-C survey, Ibrahim and Morsi (2005) conducted a discipline specific review of undergraduate and/or graduate Electrical and Computer Engineering degrees offered completely or partially online. They reviewed instructional technologies and different systems for offering electrical, electronics, and digital laboratories via distance learning to facilitate online education for engineering disciplines. They concluded that, although simulation may be used to reinforce concepts, practical experiments are needed for undergraduate electrical engineering education to develop the student's skills in dealing with the real instrumentation. They discussed if virtual labs are an alternative to the practical experience: they postulate that they should include the required hands-on control. They proposed a technology available with National Instruments LabVIEW Remote Panels, which enabled a user

to publish the front panel of a LabVIEW program for use in a standard Web browser.

Early attempts at developing remote labs were hindered by internet connectivity, hardware reliability, and the difficulty of controlling the instruments remotely with a web interface and control software. As web tools and instrument control software have become more advanced and easier to use, there has been increasing development of remote laboratories. Jodl (2007) of the Technical University of Kaiserslautern, Germany, has started an initiative for distributed remotely controlled labs (RCL) in Physics teaching. Classic physics experiments (Electron Diffraction, Photoelectrical Effect, Voltage-Current Characteristics, Diffraction and Interference) have been deployed in different European locations that can be executed through the internet. A user at a location "A" is allowed to conduct an experiment at a distant location "B" via his or her computer (client). Controlling the experiment is enabled by accessing an interface and a web server. Web cams allow the user to observe the on-going experiment. They directed these RCLs to K-12 (and as a prototype model to build-up RCLs in school projects) and to the lay public, but these remote labs could be immediately used for university teaching as well. Another system, which was introduced by Lustig et al. (2004), is known as the Internet School Experimental System (iSES). iSES has implemented seven online physics experiments. Its goal is to provide real time acquisition and remote data acquisition, data processing, and experimental control from users across the Internet.

In a recent paper, Gröber et al. (2007) review the existence and status of physics experiments in remote labs worldwide: by 2006 they found about 60 projects offering about 120 remote experiments. More than half of these projects were located in the USA and Germany, and some of the projects were joint ventures between universities in different countries.

Karadimas and Efstathiou (2007) introduce their Remote Monitored and Controlled Labora-

tory (RMCLab). RMCLab is a client-server based architecture and includes a user interface which provides capabilities such as lab administration, hardware management, and instrument operation. The primary components of their architecture include a client, an instructor-client, an application server, a resource server, and the laboratory infrastructure. Their system has been in operation since March 2004 and provides educational services to the department of electrical and computer engineering of the University of Patras, Greece.

Esche et al. (2003) developed an architecture for multi-user remote laboratories with a goal of improving engineering and science education by developing a more diverse and competitive workforce. They describe characteristics needed by remote laboratories, and these include modularity and expandability, scalability, usage of existing communication standards, and computer platform independence. Further, they posit that remote laboratories should possess the following attributes in order to be accepted by the academic community: the labs should correlate with curricular needs, should be compliant with ABET requirements, should be pedagogically sound, should be affordable, should be easy to use, and should be reliable. Esche et al.'s system is designed with the client-server network model. It provides laboratory connectivity by way of a Linux-based Web server. The server provides a graphical user interface (GUI) for the student, and it hosts a process queue along with associated input and output files for the laboratory experiments. The server is networked to individual data acquisition computers that are connected to laboratory equipment and process experiments with LabVIEW VI software. The authors claim that student performance while conducting remote laboratory experimentation is comparable to previous non-remote experimentation.

Datta and Sass (2007) introduced RBoot, which is a software infrastructure for remote field programmable gate array (FPGA) laboratories. RBoot is based on the client-server network model.

With RBoot, the student accesses the server using a secure shell (ssh) client. Then, the student uses the authors' fpga-session software to establish a lab session with one of the available ML-310 FPGA boards.

Troger et al. (2008) introduce the concept of "experiment as a service" as an extension of service oriented architectures (SOA) for virtual remote laboratories. Their architecture is built with the client-server network model and provides experimentation as services, where the experimentation service is mapped into SOA web based standards. Once a user schedules a required service, the system sends job service information to an appropriate execution host using client-server communication techniques.

Other works have established criteria for the design of remote laboratories and their associated infrastructure. Schaefer et al. (2008) provide the following desirable characteristics of remote labs:

- The user interface should be easy to understand and easy to use.
- The user (client side) should not need any special hardware or software.
- Experiments should require minimal or no interaction from on-site lab personnel once the experiment is set up.
- The system in the experiment should neither be too fast nor too slow.
- The experiment should be interesting to observe.
- The lab experiment hardware should be controllable via a computer or computer controlled equipment.
- Experiment measurements should be able to be taken via electric sensors such as ammeters, voltmeters, tachometers, electrical pressure sensors, electrical temperature sensors, etc.
- The experimental apparatus must lend itself to familiarization via videos or web tutorials.

- Access to a remote experiment is secure and protected from unauthorized users.
- Not require sophisticated communications software—it should build on standard Internet protocols
- Retain abilities for high degrees of scalability
- Provide all necessary remote laboratory management functionality
- Provide sustainability

Another criteria, as described by Salzmann and Gillet (2007), is that remote experimentation should appear to the remote student as close as possible to that of a local student. Salzmann and Gillet (2007) also enumerate challenges that will be faced when moving from small scale remote laboratory settings to large scale professional quality remote laboratory settings. They enumerate the following challenges:

- Physical equipment challenges
- Software challenges
- Maintenance challenges
- Deployment challenges
- Educational challenges
- Sustainability challenges

In this chapter, we will discuss requirements and obstacles faced by designers of remote lab infrastructures. In a recent workshop, Lindsay et al. (2007) state that remote labs are increasing in popularity, but their development thus far has been an “ad hoc” process that will not scale. The lack of a standardized infrastructure and system models for remote laboratories is one reason for the current ad hoc, non-scalable process of creating remote labs. Hence, progress towards the creation of standardized infrastructure and their system models needs to be made.

REQUIREMENTS AND CHARACTERISTICS OF SUSTAINABLE REMOTE LABORATORY INFRASTRUCTURE

In general, an infrastructure is a system of assets such as physical components, human resources, operational processes, and organizational structures required to facilitate a particular set of outcomes. For example, a country’s transportation infrastructure facilitates the delivery of raw goods, in which raw goods are used to produce products, in which products are then delivered to consumers. Naively, one might assume that the transportation infrastructure consists simply of a country’s network of roadways. However, the transportation infrastructure is more complex than just the roadway network. It consists of the roadway network system, the system of organizations producing raw goods, the system of organizations who produce products from the raw goods, the organizations who deliver the products and raw goods, and the consumers of the final product. It is easy to argue that an infrastructure is a complex “System of systems”. One particular concept common to any infrastructure is that the infrastructure’s system of assets are employed for the purpose of combining problem holders with problem solvers to produce some set of outcomes that facilitate the solution for the underlying need implied by the necessity of the infrastructure. An infrastructure is a collection (system) of assets that collectively produce a set of desired outcomes, which would not be attainable by any particular asset alone. The value added by the infrastructure is determined by the interconnection of its assets, which is the interconnection between problem holders and problem solvers. Note that problem holders and problem solvers in this context are abstract constructs.

A necessary requirement for the design of *any* infrastructure is **sustainability**. If an infrastructure cannot be sustained, then its design is flawed from conception. Therefore, when designing an infra-

structure, sustainability should drive the initial requirements for its design; it should be a “before thought”, not an “afterthought”. We propose the following (but not necessarily exhaustive) list of requirements for sustainable remote lab infrastructure (RLI). A sustainable RLI should possess the following characteristics:

- **Scalability:** An RLI must possess abilities to scale. Scalability can be described in two classes: 1) the ability to adapt to increasing demand of a particular resource and 2) the ability to increase a system’s functionality (extend its feature set) to provide new services as demand changes over time. To provide scalability, we need to provide seamless integration of assets into the RLI.
- **Manageability:** An RLI is a system of heterogeneous assets. As a result, we need mechanisms for managing a complex system of systems composed of heterogeneous assets.
- **Securable:** An RLI is inherently an information technology (IT) system. As such, we have to provide security to the system from numerous perspectives.
- **Economical:** An RLI should minimize costs while maximizing return on investment (ROI).
- **Longevity:** An RLI, once deployed, should be able to survive for long periods of time. One could argue that longevity is a result of sustainability.
- **Standardized:** The standardization process, in most cases, ensures that a system can be sustainable. The Internet is a prime example. The Internet is composed of a vast, global interconnection of independent communication network systems. Its success is a result of a standardized process used to define communication protocols between large scale, heterogeneous communication nodes.

- **Evolvable:** An RLI must be able to evolve (or adapt) to changing environments and user demand.
- **Demandable:** If there is no demand for the system, then there is no need for it.
- **Integratable:** An RLI must provide capabilities for seamless integration of new assets over time while concurrently allowing integration of legacy systems and components.
- **Modularity:** Large complex systems are more easily handled if the system contains highly modular components. From a systems engineering perspective, modularity provides understandability of the system during the design, deployment, and operational phases.

The above requirements of sustainable RLI are inter-dependent. For example, for an RLI to be evolvable, there must be a demand for new assets, and these should be integrated seamlessly, be economical, and be manageable. Longevity provides an avenue for greater ROI. Hence, the longevity requirement positively impacts the economical requirement. For this to happen, the system further requires scalability such that new assets can exist within the RLI. Management capabilities and standardization can work together to reduce costs (economic requirements) of implementing new assets when a new demand comes into existence. Modularity is a key feature that supports and positively affects scalability, evolvability, standardizability, and manageability.

Cloud computing, which is a relatively new computing paradigm, is a perfect example of a sustainable infrastructure. In fact, it is an infrastructure composed of other sustainable infrastructures including the Internet communications infrastructure, the Web (in particular, Web 2.0) infrastructure, and large-scale Enterprise server system infrastructure (ESSI). The basic idea behind cloud computing is that the “computer” lives within the “cloud”, where the cloud represents the

complex interconnection of resources required to perform computational processes. This includes anything from data storage to high performance computation. This requires the ability to interconnect the resources (the network), the resources (processing units, storage, software, etc.), and the interface into the cloud, which is normally provided via modern day web browsers using the Web 2.0 infrastructure. Cloud computing has induced the concept of “Something”-as-a-Service. The three primary services normally provided in cloud computing environments are (1) Software-as-a-Service, (2) Platform-as-a-Service, and (3) Infrastructure-as-a-Service. These three tiers of service are technologically achievable because of the ubiquity and reliability of Internet communications, advanced Web 2.0 features, and reduced cost of ESSI (due to Moore’s law) and other ancillary technologies such as software-based machine and network virtualization. Regardless, the primary feature of cloud computing is that users of cloud computing need not know anything about the cloud’s complexity. All that is required is a user’s need. The cloud computing infrastructure provides the interconnection of problem holders with problem solvers. Our thesis is that the cloud computing paradigm provides a framework in which sustainable remote laboratory infrastructures can be modeled.

When designing any complex system, a good system model is required for successful design. Cloud computing models guided the design model we propose for sustainable RLI. We have developed a distributed infrastructure with centralized interfacing (DICI) model for sustainable RLI. The system model for DICI is illustrated in Figure 1. Following our definition of infrastructure, one can notice that the DICI model contains a vast interconnection of assets. Further, a particular asset (or component) can be composed of other assets. This is driven by our desire to produce a highly modularized RLI. The goal is to have an RLI that is similar to an automated, plug-n-play

system where new assets can be integrated easily and quickly, implying economically.

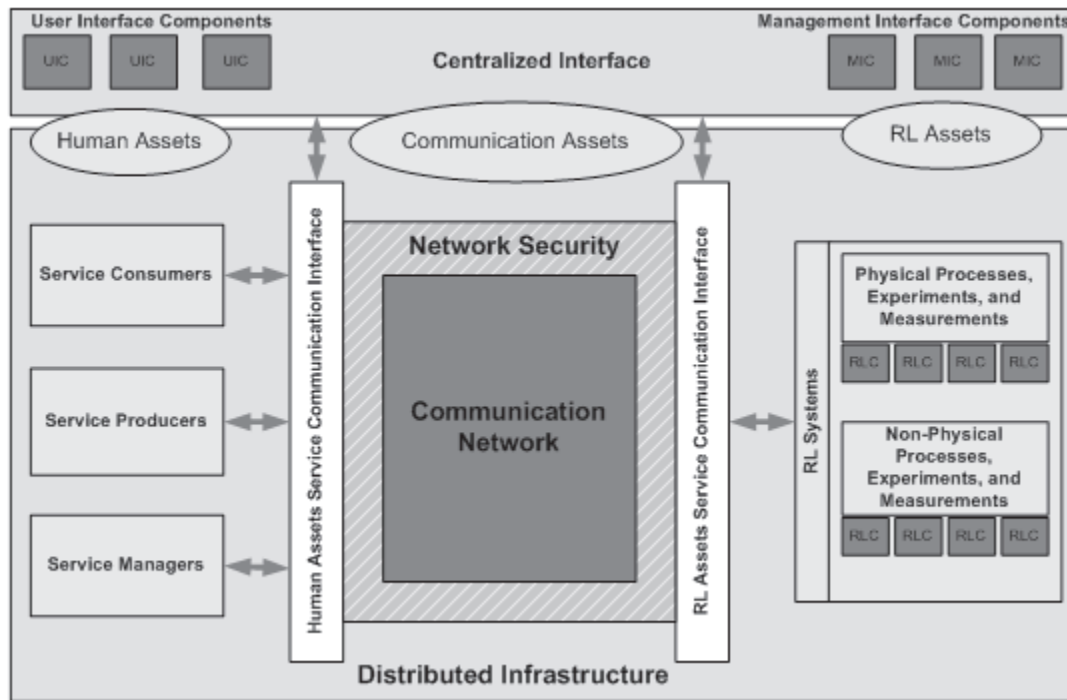
At the heart of any RLI is the communications network. This should be obvious since we are designing for remote labs, which requires Internet-based communications. The security layer that surrounds the communication framework reflects the need of security in the network and the fact that network security already exists in most university and enterprise network systems. This is an added benefit of the system, but it also causes deployment obstacles in terms of seamless integration of RLI assets. We will discuss this obstacle-benefit paradox and interesting solutions for it later in the chapter.

The DICI model incorporates concepts from the principles of software engineering design patterns. A number of reasons motivate such a design for RLI, especially as it relates to the requirements we have defined. The model provides a representation of the system’s interconnection of assets, but at a very abstract level. At the same time, it hides intricate details of complex interconnections. For example, as we will describe later, the components of the RL system can be very vast, complex, and heterogeneous. However, as the model reveals, regardless of how complex the RL system will/must be, there is a clear understanding that the complex RL system will communicate with non-RL system components over a protocol interface, which is interconnected with the centralized interface (CI). The model incorporates a coarse-grained, highly modular design, which supports adaptability and seamless integration of new assets into the RLI. Further, from the human asset perspective, DICI reveals a “service” perspective to the system user, which follows the design patterns of the cloud computing paradigm.

Detailed Description of DICI

The DICI model categorizes RLI assets into three primary groups: (1) Human Assets, (2) Communication Assets, and (3) Remote Lab Assets. Further,

Figure 1. The Distributed Infrastructure Centralized Interface (DICI) system model for sustainable RLI



human, communication, and remote lab assets are bound to both the centralized interface (CI) and the distributed infrastructure (DI). The distributed infrastructure incorporates the bulk of the RLI. However, the centralized interface, which includes two primary groups of components referred to as the user interface components (UIC) and management interface components (MIC), provides the resources that glue the system together.

Human Assets

In the DICI model, there are three types of human assets, which include Service Consumers, Service Producers, and Service Managers. Service consumers utilize the services offered by the RLI. Service consumers include, for example, students taking some class requiring a particular set of remote lab systems, researchers investigating a new design prototype, or scientists needing to observe and collect data from measurements of a

physical process. Service producers provide human resources in term of intellectual capital and labor that result in provisioning of useful services. For example, a laboratory assistant could be a service producer who installs a new set of devices and equipment into the RL system and integrates these components to form a new consumer service. Service managers administer the resources in the RLI, depending on the scope of their management roles. Service managers perform operations such as creating new user accounts, assigning user roles, scheduling laboratory courses, and scheduling system maintenance.

In the most general sense and with respect to our description of infrastructure, service producers and service managers are problem solvers, whereas service consumers are problem holders. However, note that service producers and service managers can be problem holders that seek services other service producers and service managers. Further, a particular user can simultaneously be a service

consumer, producer, and/or manager, depending on the user's role with respect to the system as a whole. For example, consider the user Alice. Alice can be a student of class A, a producer for class B, and a manager of class C. This provides an example promoting the need to categorize human assets into distinct, role-based classes.

Communication Assets

The communication assets include four primary components: (1) communication network, (2) network security, (3) human asset service communication interface (SCI), and (4) RL asset service communication interface. We assume the communication network to be based on the Internet Protocol (IP) so that standardized, ubiquitous, Internet-based communications take place. The network security component encapsulates the communication network component, which reflects the idea that securability is needed but also that in modern day enterprise network systems, it already exists in several forms, but most notably in the form of firewall systems. These firewall systems provide both a benefit and a challenge in RLI. We will discuss this issue in depth in later sections of the chapter. In order to capitalize on the ubiquitous web, the human asset SCI is specified to use web based protocols such as the Hyper-Text Transport Protocol (HTTP). Using web based protocols between human assets and the centralized interface will minimize RLI deployment costs as it removes the need to develop specialized interface software for system utilization. However, the RL asset SCI can be more diverse, and different protocols such as client-server, command and control, and peer-to-peer protocols can be used, depending on the particular requirements of a given subset of the RL system.

Remote Lab Assets

At the highest level of abstraction, RL assets are composed of RL systems. We distinguish between

two types of RL subsystems, which include physical processes, experiments, and measurements (P2EM) and non-physical processes, experiments, and measurements (NP2EM). Both P2EM and NP2EM systems include any number of remote lab components (RLC).

P2EM encompasses the category of remote labs whereby some real-world, physical phenomena needs to be measured or where some experiment studying a physical phenomena needs to take place, which usually requires capabilities of observation and data collection. For example, a P2EM remote lab could be an experiment where the effects of parameters on wave velocity in a fluid are to be studied. The remote lab could potentially include video cameras for visual observation and sensors attached at locations in the fluid tank to measure physical parameters such as velocity, acceleration, temperature, and pressure. The sensor measurements would be collected via data acquisition methods by a computer locally connected to the tank's sensor system.

NP2EM is the category of remote labs that include non-physical phenomena. This category includes environments for the purpose of emulation, simulation, and integrated development. It can also provide systems that implement "virtual labs" for enhanced learning of laboratory concepts. As an example, an NP2EM could be a remote lab where the student needs specialized software such as computer aided design/engineering tools, software compilation, software development within a specialized integrated development environment (IDE), or using specialized mathematical software. However, we can also conceive of a hybrid-remote lab where the task requires components from both P2EM and NP2EM. For example, a computer engineering student might be tasked with designing and testing a new digital to analog converter (DAC) to be used in a reconfigurable hardware device such as a field programmable gate array (FPGA). The first part of the task includes the use of a specialized IDE for the design of the DAC using a hardware description language (HDL),

synthesis of the design, and simulated timing, circuit layout, and power analysis. The second part of the task includes the use of an FPGA that implements the design in hardware and the use of signal generators, logic analyzers, and oscilloscopes for real-time, real-world testing and analysis of the design. The first part of this lab is provided by the NP2EM system services whereas the second task is provided by the P2EM system services.

CLOUDLABS 1.0: AN INSTANCE OF THE DICI SYSTEM MODEL

The DICI system model provides an architectural frame of reference that guides the design of sustainable remote laboratory infrastructures. In this section, we describe an implementation of an RLI whose design was driven by the concepts of the DICI model. We refer to our implementation as CloudLabs 1.0 (CL1). CL1 is a complete learning management system that provides a centralized mechanism for managing distributed infrastructure components such as coursework materials, audio/video resources, e-learning resources, training content, as well as management and usability functions of remote lab systems.

The centralized interface is implemented on a centralized interface server (CIS). The CIS is a central point of service consumption, production, and management for CL1. The CIS is a server system built with Web 2.0 technologies and other ancillary technologies such as structured query language (SQL) databases, storage area network (SAN) systems, and server virtualization systems. Web 2.0 offers a very rich set of advanced communication and collaboration technologies that can be utilized for RLI. Web 2.0 technologies include seamless database integration, web-based remote desktop sharing applications, real-time chat and video teleconferencing, web logs (blogs), wikis, really simple syndication (RSS, which is effectively a web-based multi-cast information distribution technology), dynamic content genera-

tion, and much, much more. An interesting feature of Web 2.0 is its ability to perform “information mashup”. Information mashup is the ability of web sites to dynamically generate web pages based on content that is relevant to the user. For example, when one logs into a Web 2.0 site, the page is generated dynamically based on information that is associated with the user’s account. This provides significant functionality for one who is developing RLI environments and UIC/MIC modules from a centralized location. In CL1, our CIS employs the Web 2.0 based Sakai open source collaboration and courseware management framework. User and Management interface components are created with the dynamic content and site builder features included within Sakai. Since CL1 uses Web 2.0 for its UICs and MICs, we are by default using the web-based, client-server communication model for the human assets service communication interface of DICI.

CL1 currently uses desktop computer lab stations that are physically connected to RLCs such as oscilloscopes, signal generators, FPGA development boards, and circuit prototyping boards. Further, software packages required by our remote lab coursework are installed on the desktop computers. We refer to these combined workstations as composite-RLCs. A composite-RLC provides to a service consumer, i.e. a student, a set of services, i.e. a particular remote lab, composed of independent RLCs that form a particular set of RLI outcomes.

CL1 implements a command and control (C&C) communications model within the remote lab asset service communication interface (RA-SCI). The C&C model is based on the idea of service nodes within the network periodically sending command requests (polling) to a command and control server (CCS). When a CCS receives a command request from a service node, it will answer the service node with a command response. Once the service node receives the command response, it will perform tasks as defined by the response. The C&C model provides interesting

features for seamless integration and management of resources in RLI. When deploying an RLI with large numbers of RLC resources, several challenges will need to be addressed. Some of these challenges include resource management, resource scheduling, resource discovery and registration, resource service binding, and resource configuration. Another challenge faced with deploying RLC resources in large-scale is a result of the network security component of the communication assets. Modern enterprise network systems such as in university and industrial campuses, which comprise the primary entities that will deploy RLIs, have complex networking systems, and these complex networking systems are overlaid with complex firewall systems. As such, there is significant overhead when deploying network services within these networks, especially in terms of large scale deployment. To provide the reader with a deeper understanding of this complexity issue within RLI, we provide a brief background discussion on firewall systems and the network security change management process.

Firewall Systems and the Network Security Change Management Process

To understand the underlying issues with enterprise network security and network security change management, one needs to have an understanding of firewall technology.

Firewalls and Network Security

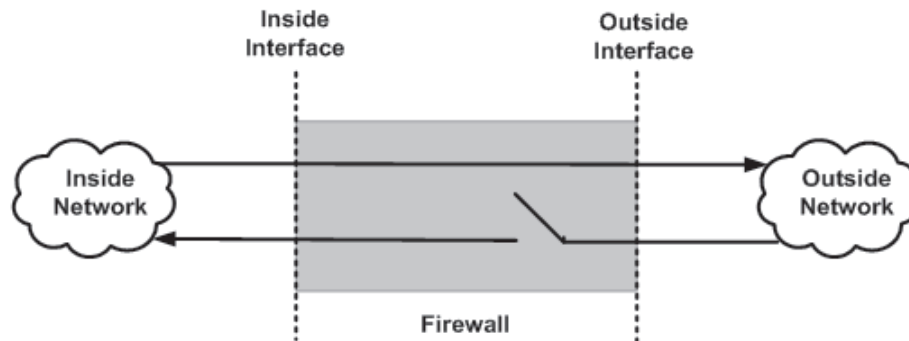
The Internet was originally designed as an “open” communications architecture. However, as the Internet grew and became commercialized, network security threats caused network administrators across the Internet to “close” their networks with firewalls. Firewalls are network devices that filter incoming and outgoing network traffic (computer communication connections) based on a network security policy, and they are commonly used to par-

tition enterprise networks into secure regions. All traffic flowing between the regions will traverse the firewall. For each communication connection traversing the firewall, the firewall will use data contained within the communication unit (a network data packet) and compare the data against the firewall’s security policy, which is a set of packet filter statements. If the policy is defined to “allow” a particular type of traffic, then the firewall will provision the traffic and allow it to continue its flow. Otherwise, the firewall will filter the traffic and not allow (deny) it to continue flowing—i.e., it prevents the traversal of the communication connection through the firewall.

Figure 2 provides a visual representation of a firewall. When designing firewall systems, the designer must understand the topology of the network and the firewall’s orientation with respect to the topology. The idea of inside and outside interfaces is tightly coupled with the topological perspective. From Figure 2, the firewall provides a secure network on the inside interface while the outside interface connects to an external network—external with respect to the firewall and the network topology. Communication traffic flows from the inside network through the firewall to the outside network and vice versa. Common enterprise networks and their firewall systems normally allow most traffic to flow freely from the inside to the outside. However, the firewall acts as a modulated switch for traffic that is initiated from the outside to nodes (computers) on the inside. The “modulation” is provided by a security policy. An example security policy could be: “Allow all Internet connectivity to the web server located within the inside network.” This policy is implemented via a firewall filtering rule, which is a translation of the high-level security policy into a filtering language understood by the firewall. The “switch” closes for traffic that matches the firewall’s security policy and opens for traffic that does not match the policy.

There are two types of firewall security policies: dynamic and static. Static policies are the

Figure 2. An abstract visualization of a firewall



formal policies defined by the enterprise. Dynamic policies are policies created by the firewall when an inside node initiates a communications connection to an outside node. Notice that when an inside node communicates with an outside node, the outside node will reply to the inside node. Hence, there will be traffic flowing from the outside to the inside. In order to allow reply connections to the internally initiated connections, the firewall employs stateful packet filtering technology. With stateful filtering, the firewall stores connection tracking state information for every internally initiated connection. Once the external node replies to the internal node, the firewall will inspect the connection tracking state table. If the reply connection matches information stored in the state table, then the connection is allowed to pass through the firewall.

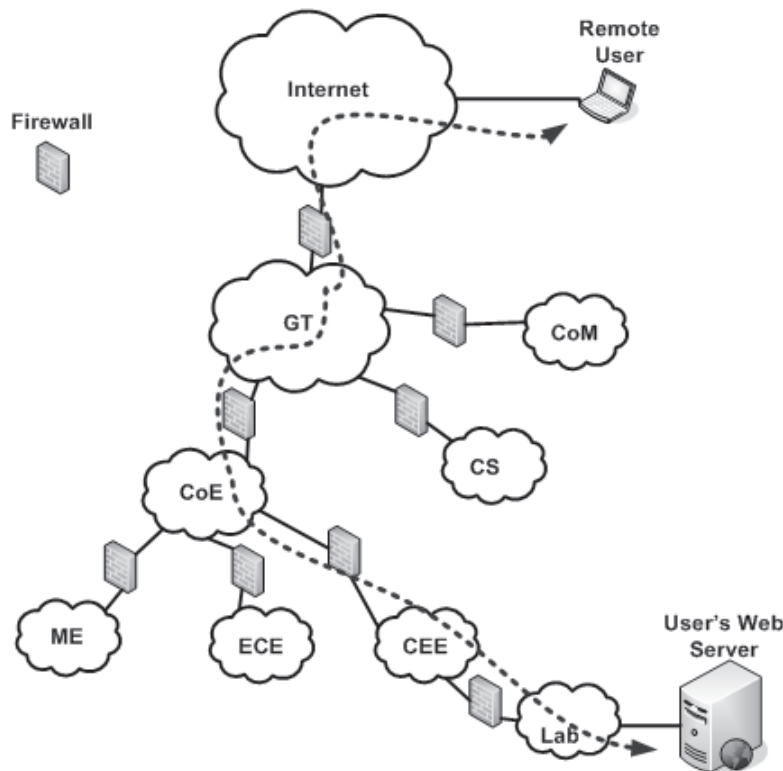
The Network Security Change Management Process

We now provide an example of the enterprise network security change management process in order to show the complexities that will be involved with large scale RLI deployments. Figure 3 illustrates an enterprise network topology. Note that the network is partitioned into various network regions by firewalls. In this case, the network regions correspond to different enterprise units. For example, the top level region named “GT” is

separated by the second level regions such as CoE (college of engineering), CS (computer science department) and CoM (college of management). CoE is further divided into the following units: ME (mechanical engineering), ECE (electrical and computer engineering), and CEE (civil and environmental engineering). Finally, CEE has a separate lab region. Consider the case where a user in CEE’s lab region wants to add a web server to the network and needs the “world” (i.e. anyone who can connect to the Internet) to be able to retrieve web pages served by the web server. This is represented by the dotted line in Figure 3. All of the firewalls on this network path must be configured with a security policy that allows network traffic that is destined to the user’s web server. For this to happen, a collaborative effort between the user, system administrators, network administrators, and security administrators (and possibly others) will be required. The following is a common set of steps required for this change management process:

- The user establishes a networking and/or security need
- The user communicates this need to a system administrator
- The system administrator works with user and then establishes the need with the security administrator

Figure 3. Example of an enterprise network topology and its overlaid firewall system



- The security administrator works with the system administrator and then establishes the need with the network administrator
- The network administrator implements the final process to solve the need

First, the user establishes a need, which in this case, is to make his/her web server available to the world. Second, the user communicates this need to a system administrator. Third, the system administrator installs the required software, configures the system, and applies any necessary security updates to the system.

Fourth, the system administrator communicates the need to the security administrator. Fifth, the security administrator will perform vulnerability and risk assessment processes against the system (the web server) to ensure that it meets the enterprise's required security standards. This step

is normally an iterative process between the security administrator and system administrator—i.e. the security administrator will communicate any security issues to the system administrator. Then, the system administrator will apply appropriate software/hardware updates to the system according to the security administrator's advice. This part of the process stops once the system meets the security standards and receives a "clean bill of health". Sixth, the security administrator will coordinate with network administrators so that appropriate security policies are updated on the necessary firewalls and other network security devices along the network path (the path illustrated by the dotted line in Figure 3) to the web server. Once the security policies have been updated, the change management process is complete and the system is available for users on the Internet.

Considering that RLI environments will contain many devices (RLCs) that require remote accessibility, one can see that the network security change management process will require significant amounts of resources for large scale RLC integration. For example, anytime a new RLC is added to or taken from the remote laboratory or moved from one network location to another, the change management process will need to be invoked. This is further compounded if the RLC (or system of RLCs) is moved from one enterprise to another (for example, if the lab is mobile and used at multiple universities). In this case, the change management process must be a collaborative effort across multiple enterprise systems. This complicated task prevents seamless integration and increases system costs. This leads to the following questions: “How can remote labs be implemented on a large scale without requiring the costly change management process to be invoked?” and “How can remote labs retain high-levels of security?”

Another question revolves around the issue of RLC connectivity. In Internet environments, computer to computer communication is done with Internet Protocol (IP) addresses. IP addresses are very dynamic. The addresses change if the node moves between different networks. Further, IP addresses change due to dynamic host configuration protocol (DHCP) and network address translation (NAT). So, another question is “How do we manage dynamic IP addresses within the RLI, especially when dealing with large numbers of RLCs and how do we distribute the connectivity information to the remote users in a seamless manner?” As we will see, CL1 utilizes the command and control communications model to alleviate this integration and connectivity obstacle in RLI.

CloudLabs 1.0 Infrastructure

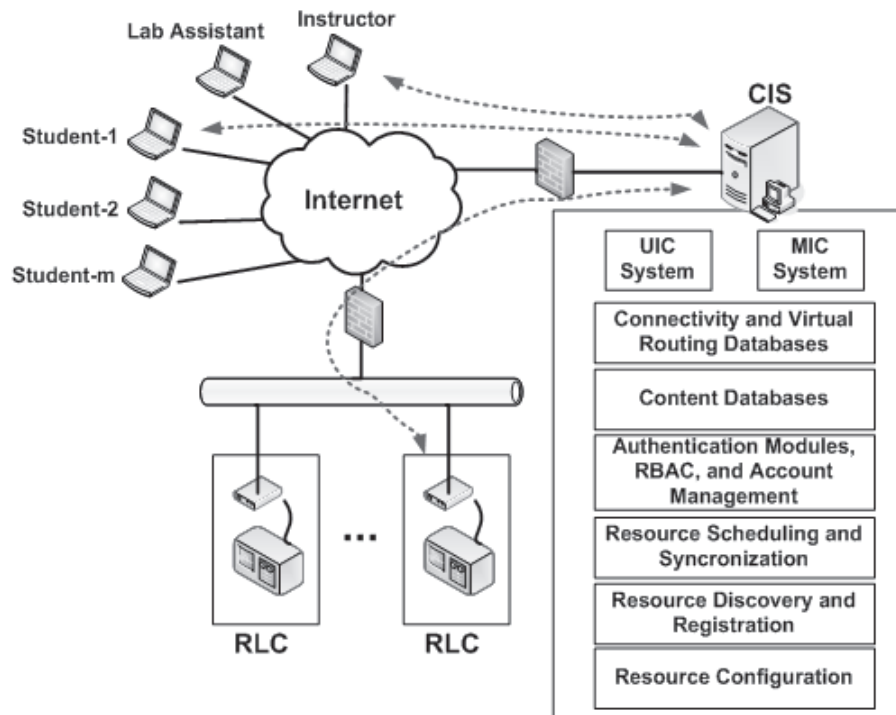
The CloudLabs 1.0 infrastructure contains both distributed infrastructure components and centralized interfacing components, as guided by

the DIC1 model. Figure 4 provides a high-level illustration of CL1’s architecture.

In CL1, the centralized interface is implemented within a centralized interface server (CIS) that uses the Web 2.0 based Sakai software system. The user interface component is dynamically generated based on the user’s role when entering the system. Recall that a user can be a service consumer, producer, and/or manager. The content provided to the user via the Web 2.0 interface is generated dynamically, depending on the service context of the system that the user requests at any given time. This feature is implemented with technology known as role-based access control (RBAC). Consider, for example, a user Alice who is a student of class A, an assistant of class B, and a manager of class C. When Alice selects services and features of class A, the system dynamically retrieves her role and its associated access control properties assigned to her for class A. Based on the access control properties and class context, the system dynamically generates the content associated with class A. This content could include features such as a class wiki, chat room, video teleconference application module, and remote lab connectivity links. Assume further that at some time after Alice has accessed class A content, she needs to perform tasks related to her job as a lab assistance for class B. Once she selects the features of class B, the system retrieves her role as a service producer of class B, applies appropriate access controls, and dynamically generates class B content.

The subcomponents comprising the UIC and MIC systems include connectivity and virtual routing databases, content databases, authentication modules, RBAC, account management, and resource scheduling, synchronization, discovery, registration, and configuration. Implementing and managing these features in a centralized manner over Web 2.0 technology provides capabilities meeting the requirements of sustainable RLI. Detailed explanation of the CL1

Figure 4. The CloudLabs 1.0 infrastructure: An instance of the DIC1 model



resources will be explained in the case study of the following section.

CASE STUDY: A CLOUDLABS 1.0 DIGITAL DESIGN REMOTE LAB

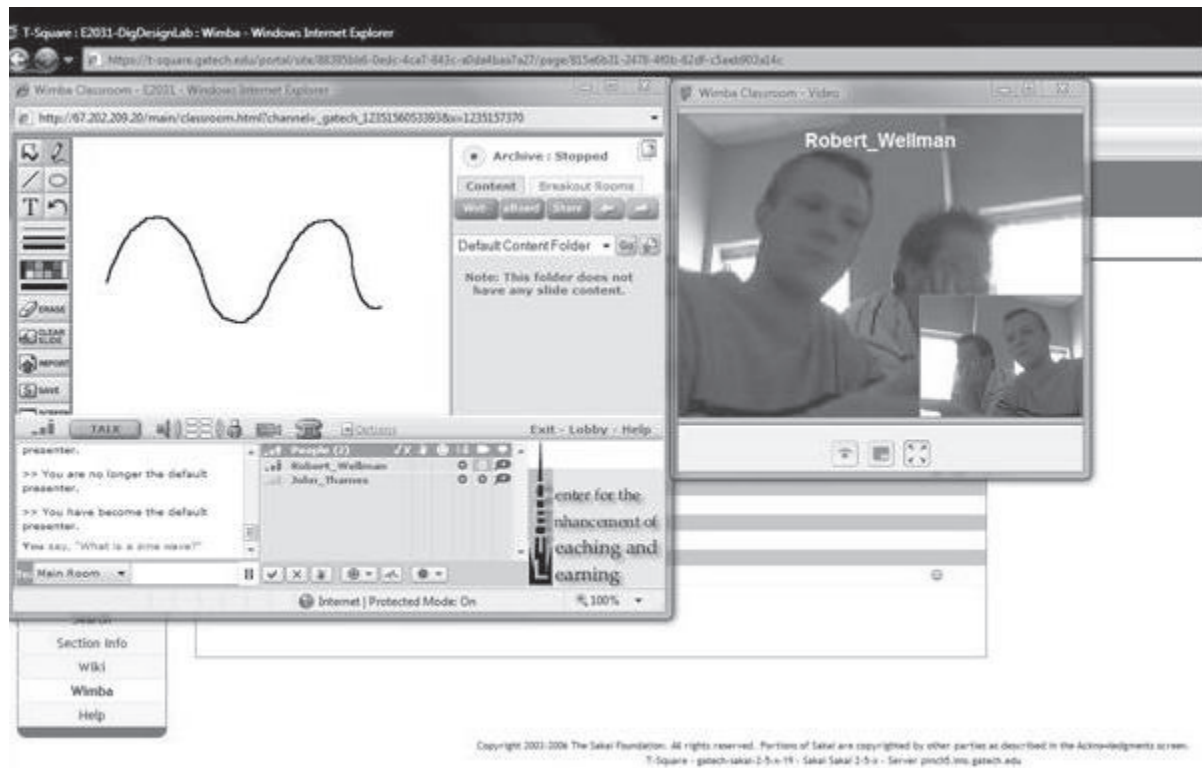
We now provide a case study of the CloudLabs 1.0 infrastructure that describes a real-world remote laboratory instance and elaborates on design details of the architecture. In this case study, remote students are attending an E2031 Digital Design Laboratory, and the students are preparing to execute their first laboratory assignment, which is an exercise in signal measurement.

To begin the remote lab, the students attend a synchronous pre-lab session with the lab instructor. The students and instructor login to the CIS web interface at a pre-determined time where a web-based collaboration tool, which is a user interface component of the CIS, is used to dis-

cuss the pre-lab topics. The pre-lab collaboration tool, shown in Figure 5, provides features such as electronic whiteboard, video and audio interaction, and remote desktop sharing capabilities. In Figure 5, the instructor is communicating with the remote students using real-time audio and video for a discussion of sinusoidal signals. Further, the instructor can share his desktop applications to the remote students. For example, the instructor might plot a signal using a software tool and then share the plot with the remote students.

Once the pre-lab session has ended, the student will execute the signal measurement laboratory exercises from within another CIS user interface component (UIC) during his/her scheduled time frame, unless the lab is done on a first-come, first-serve basis. In either case, the CIS management interface components (MIC), such as the scheduling and virtual routing components, will manage, on demand, the user's ability to connect to an available RLC resource. In this case study,

Figure 5. E2031 pre-lab collaboration interface



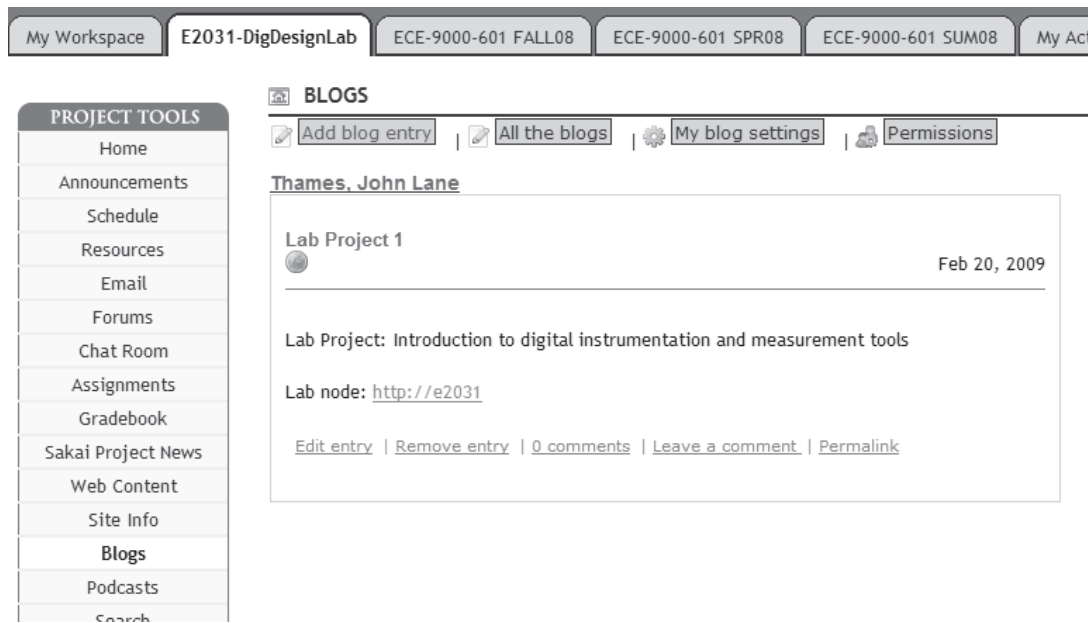
the RLC resources are the composite-RLC desk-top lab stations as described previously.

The student will access the RLC resource using the interface shown in Figure 6. In this example, the student chooses the E2031-DigDesignLab tab. Once the student clicks the pseudo-hyperlink labeled "Lab node: <http://e2031>", which is located in the middle of the web page of Figure 6, the appropriate MICs within the CIS will establish virtual connectivity between the student's UIC and some available RLC resource, which has been assigned to this class via the scheduling MIC. We refer to this linkage label as a pseudo-hyperlink because it is not a hyperlink in the traditional sense. The pseudo-hyperlink, once clicked, activates a connectivity event within the underlying CIS MIC system. The connectivity event causes a query into the virtual routing, scheduling, and connectivity

subsystems. If appropriate RLC resources are available, as determined by the results of the connectivity event query, then a virtual route between the student's UIC and the RLC resource(s) will be established. We refer to this as a virtual route because it is not a route in the classical sense of networking. The virtual route in the CL1 context refers to a process of transferring communication packets between the RLC and the UIC. To establish a virtual route, the CIS sends a command response to the appropriate RLC upon the RLC's next poll. The command response will depend on the type of resource service being requested via the student. Once the RLC receives the command response, it will start the appropriate processes and communications as defined by the response protocol.

We will use the network timing diagram shown in Figure 7 to provide the reader with a deeper

Figure 6. Student interface for establishing communication session with RLC



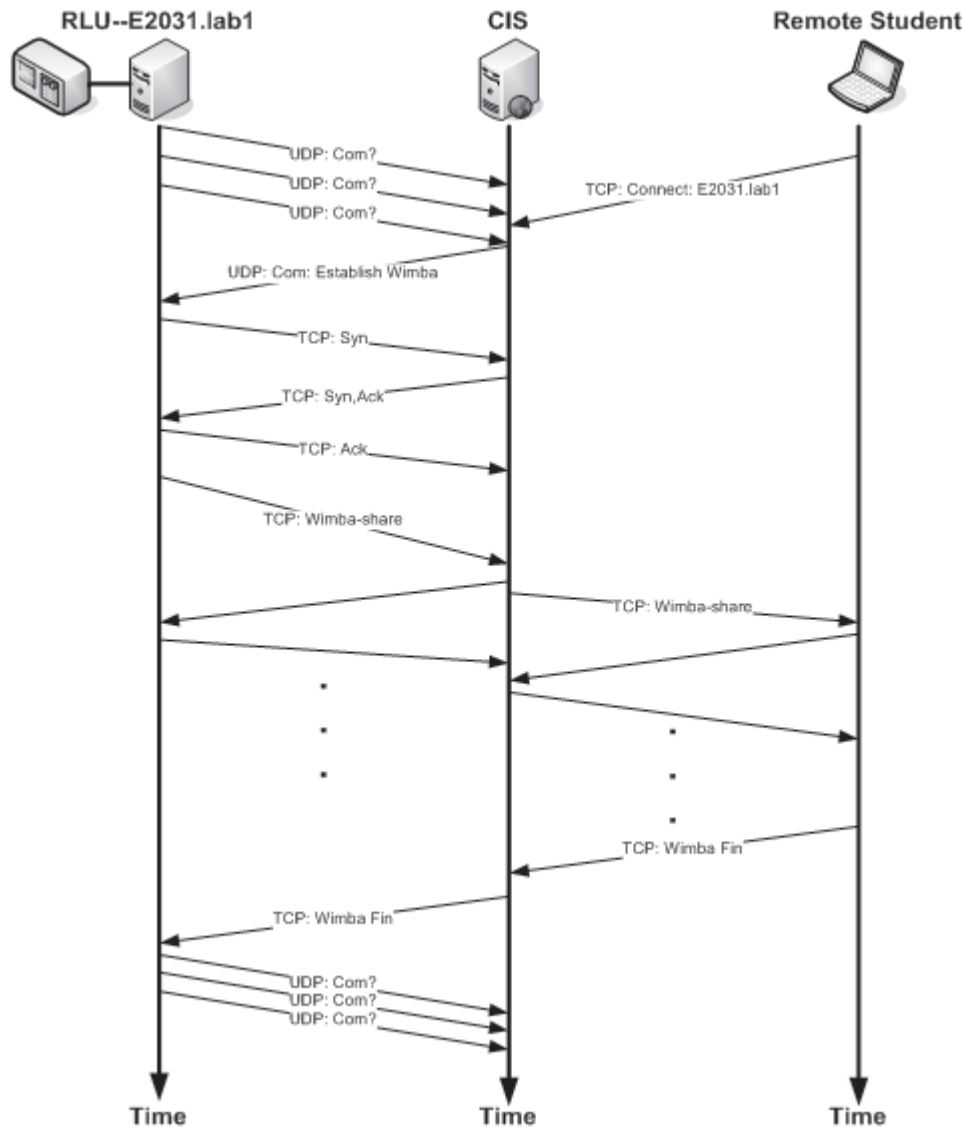
understanding of the underlying C&C communications process used in CL1. In Figure 7, the vertical axes represent time relative to each network node with time increasing in the downward direction. The angular lines connecting one node's time axis to another represents a communication flow. Note that the angular line starts from one node and terminates at the other node's axis at some delta-time later. This represents the time delay inherent in communication networks. For example, if the RLC sends a communication to the CIS at time t , then the CIS will receive the connection at some time $t + \Delta t$ later. The angular lines are labeled with information representing the type of communication for that particular flow. For example, the top three lines from the RLC to the CIS represent the polling connections. In this example, the RLC is communicating with the CIS via the user datagram protocol (UDP). For each one of these three connections, the RLC is asking the CIS if it needs to execute any new commands.

As an aside, recall that one of the communication complexity issues associated with RLI and the integration and management of its RLC com-

ponents was concerned with dynamic IP addresses. So, how does the C&C communication model in CL1 solve this? This is solved via discovery and registration processes. The CIS's IP address is assumed to be non-dynamic. In other words, the CIS does not relocate to new networks (non-mobile), does not use DHCP, and does not use dynamic NAT. The RLC's command-control software will monitor its own IP address. If the RLC's IP address changes, it will send a registration command to the CIS and inform it of the RLC's new IP address (discovery). The CIS will then update the MIC database systems with the new connectivity information.

We now continue with the original discussion. At some point in time between the RLC's second and third polling interval, the remote student clicks the pseudo-hyperlink in the UIC (as shown in Figure 6). At this point, the CIS knows that the student needs to access an RLC within the E2031 RL subsystem. During the chosen RLC's next poll, the CIS sends a reply command back to the RLC telling it to establish a transport control protocol (TCP) session with the CIS's Wimba UIC Web

Figure 7. Timing diagram representing student, CIS, and RLC communication streams

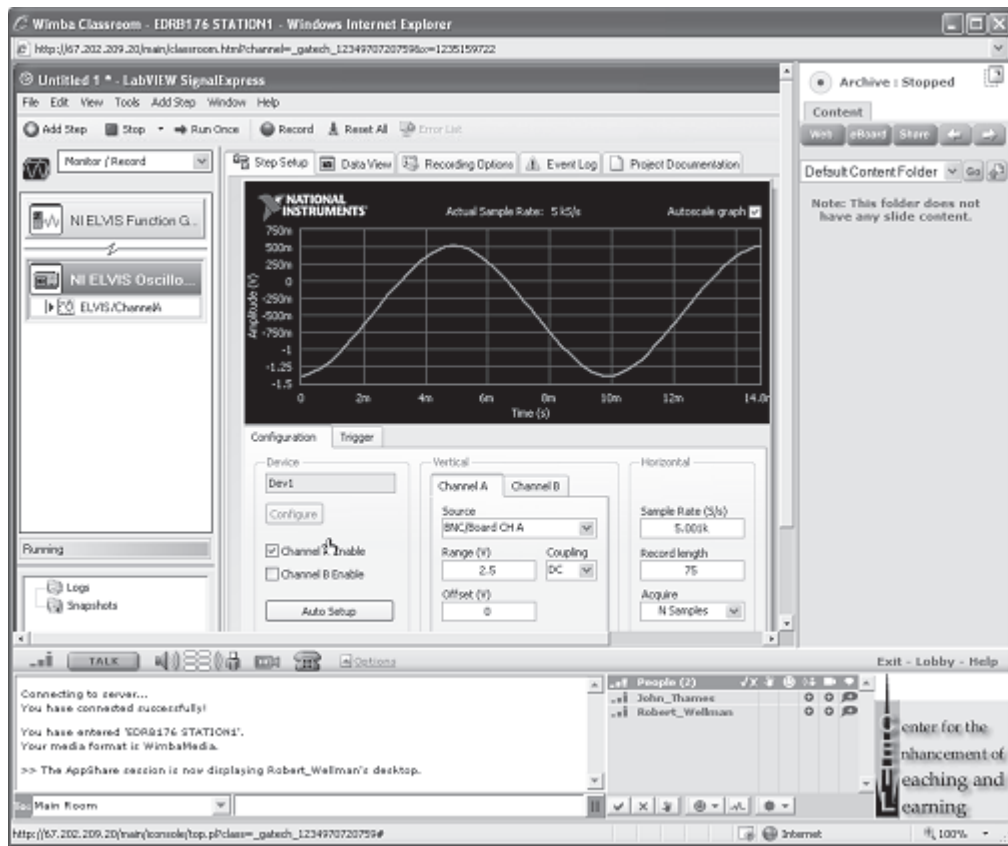


2.0 interface. (Wimba is Web 2.0 collaboration software that has been deployed within CL1). Once the RLC receives the connection establishment command, it issues the TCP “3-way handshake” to establish a logical TCP circuit with the Wimba UIC. This is shown with the TCP:SYN, TCP:SYN/ACK, and TCP:ACK connection streams between the RLC and the CIS in Figure 8. Once the 3-way handshake is complete, the CIS interconnects the student’s UIC with the RLC’s

Wimba UIC via virtual routing. At this point in time, the student has a virtual connection with the RLC, and the student can operate the remote desktop lab station just as if he/she were to login directly to the station using, say, remote desktop application software.

Figure 8 shows the UIC from the student’s perspective. Note that the remote desktop session is embedded within the Wimba web interface. From this point, the student can perform all of the

Figure 8. RLC desktop session as seen by student from CIS web interface



laboratory exercises from within this interface. Once the remote lab exercises are complete, the student will close the Wimba UIC session and the CIS will instruct the RLC to terminate its Wimba connection and return to its polling state, which is revealed in the timing diagram as the TCP:Wimba Fin connection. The RLC will tear down the Wimba session once it receives the TCP:Wimba Fin command. After the session is terminated, the RLC resumes its command request polling connections, as illustrated by the bottom three connection streams in the timing diagram.

There are several benefits for this type of infrastructure. First, there is a single point (the CIS) that enables user and management functionality. Second, the CIS can implement authentication and role-based access control in an automated and seamless fashion. Third, the network secu-

rity change management process does not need to be invoked for any RLC device. The change management process will happen only once, and that occurs when the CIS is first placed on the network. Fourth, since the RLC devices are not accessible through the firewall to the Internet and only accept command instructions from the CIS, and since the CIS is providing authentication and role-based access control, the architecture retains a very high-level of remote laboratory security. Finally, the UIC and MIC components within the CIS can be implemented with Web 2.0 technology such as the ones described in this section. Obviously, these benefits coincide with the requirements of sustainable RLI.

FUTURE TRENDS

CloudLabs 1.0 provides an interesting framework for remote laboratory infrastructure. Building on the foundations of CL1, we can envision several extensions to CL1 to move towards the next generation of CL1, progressing it towards CloudLabs 2.0. The framework encompasses the ideas of “Something”-as-a-Service from the cloud computing paradigm. Future RL components could be extremely modular. For example, consider the case where one has an oscilloscope in London measuring signals produced from a circuit in Russia. This might sound far-fetched. But, with advances in sensor systems/networks and digital signal processing, such a feat could be achievable for certain circumstances. This could give rise to the idea of Components-as-a-Service in the CloudLabs infrastructure. Other technologies that can be employed with the infrastructure include augmented reality, real-time remote lab session archival, and agent-based knowledge extraction from archived laboratory sessions. With a modular infrastructure based on a sound system model, the incorporation of diverse technological features can exist.

CONCLUSION

The need for remote laboratories in higher education is a recent demand resulting from wide-scale adoption of distance learning technology, all of which being driven by the ubiquitous Internet and the ubiquitous Web. The focus of this chapter was centered on the requirements and design for remote laboratories and their necessary infrastructure. Requirements and characteristics of sustainable remote laboratories were defined, and the distributed infrastructure and centralized interface (DICI) system model was introduced. Based on the DICI system model, the authors have implemented and studied a remote laboratory infrastructure, CloudLabs 1.0, that provides

a highly modular, easily extendible, seamlessly integratable, and economical infrastructure for remote laboratories. Continued integration of new technology and feature sets into the CloudLabs infrastructure shows promise for providing a very flexible and enhanced learning management system for both educators and students involved with distance learning of the future.

REFERENCES

- Datta, K., & Sass, R. (2007). RBoot: Software infrastructure for a remote FPGA laboratory. In *15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines* (pp. 343-344). IEEE FCCM.
- Enloe, C., Pakula, W., Finney, G., & Haaland, R. (1999). Teleoperation in the undergraduate physics laboratory-Teaching an old dog new tricks. *IEEE Transactions on Education*, 42(3), 174-179. doi:10.1109/13.779895
- Esche, S., Chassapis, C., Nazalewicz, J., & Hromin, D. (2003). An architecture for multi-user remote laboratories. *World Transactions on Engineering and Technology Education*, 2(1), 7-11.
- Gröber, S., Vetter, M., Eckert, B., & Jodl, H. (2007). Experimenting from a distance-Remotely controlled laboratory. *European Journal of Physics*, 28(3), 127-150. doi:10.1088/0143-0807/28/3/S12
- Hamza, K., Alhalabi, B., & Marcovitz, D. (2000). Remote labs. In C. Crawford, et al. (Eds.), *The Society for Information Technology and Teacher Education International Conference*, (pp. 155-161).
- Henry, J., & Knight, C. (2000). Improving laboratories with internet controlled equipment and student support. In *ASEE Southeastern Section Conference*, (pp. 1-8). ASEE.

- Ibrahim, W., & Morsi, R. (2005). Online engineering education: A comprehensive review. In *American Society for Engineering Education Annual Conference & Exposition*, (pp. 1010-1023). ASEE.
- Jodl, H. (2007). *RCL-Remotely controlled laboratories*. Retrieved August 10, 2010, from <http://rcl.physik.uni-kl.de/>
- Karadimas, D., & Efstathiou, K. (2007). An integrated educational platform implementing real, remote lab-experiments for electrical engineering courses. *Journal of Computers*, 2(2), 37–44. doi:10.4304/jcp.2.2.37-44
- Kirkland, N., Vitanov, V., & Schaefer, D. (2008). An investigation into using current information technologies to provide engineering education to sub-Saharan Africa. *International Journal of Engineering Education*, 24(2), 345–356.
- Lindsay, E., Long, P., & Imbrie, P. (2007). Workshop-remote laboratories: Approaches for the future. In *37th ASEE/IEEE Frontiers in Education Conference*, (pp. W1C1-W1C2). ASEE/IEEE
- Lustig, F., Schauer, F., & Ozvoldova, M. (2004). The internet school experimental system-iSES. Retrieved November 13, 2010, from <http://www.ises.info>
- Rippel, M., Panchal, J., Schaefer, D., & Mistree, F. (in press). Fostering collaborative learning and educational mass customization in a graduate level engineering design course. *International Journal of Engineering Education*, 25(4).
- Salzmann, C., & Gillet, D. (2007). Challenges in remote laboratory sustainability. In *International Conference on Engineering Education*, (pp. 1-6). ICEE.
- Schaefer, D., Scott, D., Molina, G., Al-Kalaani, Y., Murphy, T., Johnson, W., & Goesser, P. (2008). Integration of distance learning technology into traditional engineering physical laboratory exercises. In *ASEE Southeast Section Conference*, (pp. 1-10). ASEE.
- Shen, H., Xu, Z., Dalager, B., Kristiansen, V., Strøm, O., & Shur, M. (1999). Conducting laboratory experiments over the internet. *IEEE Transactions on Education*, 42(3), 180–185. doi:10.1109/13.779896
- Sloan-C. (2004). Entering the mainstream: The quality and extent of online education in the United States. *Sloan-C and the Sloan Center for OnLine Education (SCOLE)*, 1-27. Retrieved June 19, 2010, from http://sloanconsortium.org/resources/entering_mainstream.pdf
- Troger, P., Rasche, A., Feinbube, F., & Wierschke, R. (2008). SOA meets robots-A service based software infrastructure for remote laboratories. *International Journal of Online Engineering*, 4(2), 57–62.

ADDITIONAL READING

- Chang, G., Yeh, Z., Chang, H., & Pan, S. (2005). Teaching photonics laboratory using remote-control web technologies. *IEEE Transactions on Education*, 48(4), 642–651. doi:10.1109/TE.2005.850716
- Gurkan, D., Mickelson, A., & Benhaddou, D. (2008). Remote laboratories for optical circuits. *IEEE Transactions on Education*, 51(1), 53–60. doi:10.1109/TE.2007.900018
- Kikuchi, T., Fukada, S., Fukuzaki, A., Nagaoka, K., Kenjo, T., & Harris, D. (2004). DVTS-based remote laboratory across the Pacific over the gigabit network. *IEEE Transactions on Education*, 47(1), 26–32. doi:10.1109/TE.2003.816067

Lang, D., Mengelkamp, C., Jager, R., Geoffroy, D., Billaud, M., & Zimmer, T. (2007). Pedagogical evaluation of remote laboratories in emerge project. *European Journal of Engineering Education*, 32(1), 55–72. doi:10.1080/03043790601055626

Marques, R. (2008). Design and implementation of a reconfigurable remote laboratory using oscilloscope and PLC network for WWW access. *IEEE Transactions on Industrial Electronics*, 55(6), 2425–2432. doi:10.1109/TIE.2008.922400

Nickerson, J., Corter, J., Esche, S., & Chassapis, C. (2007). A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Journal of Computers and Education*, 49(3), 708–725. doi:10.1016/j.compedu.2005.11.019

Ogot, M., Elliott, G., & Glumac, N. (2003). An assessment of in-person and remotely operated laboratories. *Journal of Engineering Education*, 92(1), 57–63.

Sanchez, J., Dormido, S., Pastor, R., & Morilla, F. (2004). A java/matlab-based environment for remote control system laboratories: Illustrated with an inverted pendulum. *IEEE Transactions on Education*, 47(3), 321–329. doi:10.1109/TE.2004.825525

Scheets, G., Weisner, M., & Sharda, R. (2005). Changing a standard telecommunications laboratory to a same-time-different-place virtual laboratory format: Techniques utilized and lessons learned. *IEEE Transactions on Education*, 48(4), 713–718. doi:10.1109/TE.2005.850717

Sebastian, J., Garcia, D., & Sanchez, F. (2003). Remote-access education based on image acquisition and processing through the Internet. *IEEE Transactions on Education*, 46(1), 142–148. doi:10.1109/TE.2002.808237

Stancil, D., Gist, N., & Jiang, Y. (2007). REAL: The remote educational antenna laboratory. In *IEEE International Symposium on Antennas and Propagation* (pp. 5399–5402). IEEE.

Tzafestas, C., Palaiologou, N., & Alifragis, M. (2006). Virtual and remote robotic laboratory: Comparative experimental evaluation. *IEEE Transactions on Education*, 49(3), 360–369. doi:10.1109/TE.2006.879255

KEY TERMS AND DEFINITIONS

Client-Server Communication Model: A communication model whereby two nodes communicate with each other and one node, the server, provides a network service to any number of client communication nodes. Examples include web, email, and file servers.

Cloud Computing Paradigm: A computing paradigm whereby traditional computational resources such as processing and storage are maintained within the network and are available as services to users of the cloud computing system.

Command and Control Communications Model: A communication model whereby a communicating node queries a centralized command and control node asking for command responses.